## REMARKS

Applicant is in receipt of the Office Action mailed October 22, 2003. Claims 1-81 remain pending in the case. Further consideration of the present case is earnestly requested in light of the following remarks.

### Section 103 Rejections

Claims 1-81 were rejected under 35 U.S.C. 103(a) as being unpatentable over Volk et al (U.S. Patent 5673401, "Volk") and Gipalo (U.S. Patent 6348934, "Gipalo"). Applicant respectfully disagrees.

Independent claim 1 recites:

1. A computer-implemented method for programmatically modifying a graphical program, comprising:

executing a graphical program generation (GPG) program;

the GPG program receiving information, wherein the information specifies functionality of the graphical program;

the GPG program programmatically modifying the graphical program in response to said information specifying the functionality of the graphical program, such that the graphical program implements the specified functionality.

As the Examiner is certainly aware, to establish a prima facie obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. In re Bond, 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). As held by the U.S. Court of Appeals for the Federal Circuit in Ecolochem Inc. v. Southern California Edison Co., an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis.

In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular . . .. Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

Applicant submits that none of the cited art provides a motivation to combine. Rather, the Examiner has simply asserted that an improvement would result from the combination, specifically, that "it would be a convenient way to effectively modifying [sic] and storing [sic] the changes without the user being involved in the inner workings of the software." Applicant submits that not only is there no motivation to combine in the cited art, but that even if the references were combined, the resulting combination would not produce Applicant's invention as claimed.

Regarding claim 1, the Office Action asserts that Volk shows the method for modifying a graphical program, including executing a graphical program (citing Figure 1, 16A-B, and column 5, lines 40-55), the program receiving functionality information and modifying the graphical program to implement the specified functionality (citing column 5, lines 30-45, column 6, lines 9-17 and lines 35-60, and column 10, lines 15-38).

Applicant notes that in the present application, a graphical program is defined as *a plurality of interconnected nodes that visually indicate functionality of the graphical program*. Applicant respectfully submits that Volk does not teach a graphical program as defined in the present application, and in fact does not suggest or even mention a graphical program at all. Rather, as stated in the title, Volk discloses a system and method for *a customizable sprite-based graphical user interface*. More specifically, Volk discloses a user interface for an interactive television system. Applicant notes that graphical user interfaces are quite different from graphical programs.

The cited Figure 1 actually illustrates an operating environment for an interactive network system, as stated in col. 11, lines 9-11. Cited Figures 16A-16B are logical flow diagrams that illustrate steps with a computer-implemented routine 1600 that is responsible for the display of the screen 95 in a graphical viewer interface 100. In other words, Figures 16A and 16B flowchart a method for creating/displaying a graphical user interface, and specifically do not describe a graphical program.

Similarly column 5, lines 40-55 describes *a method of displaying a control item on a display screen and for graphically manipulating user controls imparted to the control item via a user input device.* Again, the cited art pertains to a graphical user interface, and does not teach or suggest a graphical program as defined in the present application.

Additionally, column 5, lines 30-45, column 6, lines 9-17 and lines 35-60, and column 10, lines 15-38 provide further descriptions of a graphical user interface, not a graphical program. For example, column 5, lines 30-45 states that typical control items include *buttons for executing a command for indicating a preference, such as a push button for executing a single specific command; and lists for allowing a user to select an item from a list of items, including spin dials and spin lists.* Column 6, lines 9-17 describes assembling a control item (i.e., a graphical user interface item) from control elements or interface components, such as *a shape item, a text item, a picture item, a sound item, a list item, a video item, or an alignment item.* Again, this passage refers to construction of a graphical user interface item, and does not describe a graphical program. Finally, column 10, lines 15-38 describe graphical indicators of user interface "focus" for control items in the interface.

Thus, Applicant respectfully submits that Volk neither teaches nor suggests a graphical program, and specifically does not teach or suggest executing a graphical program generation (GPG) program, the GPG program receiving information, wherein the information specifies functionality of the graphical program, and the GPG program programmatically modifying the graphical program in response to said information specifying the functionality of the graphical program, such that the graphical program implements the specified functionality.

The Office Action also asserts that Gipalo shows the program generation program aspect of claim 1, and further asserts that combining Gipalo with Volk produces Applicant's invention as represented in claim 1. Applicant respectfully disagrees.

As stated in column 2, lines 28-32, Gipalo's invention *uses generic statements in the commands in the underlying computer program that reference a series of tables, in which are stored the values necessary to effect the desired command*, [in order] *to enable rapid modification of the program that controls a graphical user interface.*

The cited passages of Gipalo describe *processing a method for processing information via a graphical user interface in a client-server environment, in which data displayed on the client can be rapidly changed to accommodate changing user requirements.* More specifically, the cited passages describe the use of "Meta Data Tables" to control and configure a graphical user interface (GUI), for example, by conversion between English commands used in the GUI and the more cryptic coded language that implements or specifies the actual performed command, i.e., by translating the cryptic code to English. As another example, Gipalo describes using the Meta Data Table to *define the nodes within the tree of screens presented to the user in a Graphical User Interface (GUI). By defining the nodes in the tree in a Meta Data Table, these nodes can be changed easily, but even more significant, the underlying Visual Basic code can be significantly reduced in size.* Finally, in a more general sense, Gipalo characterizes the invention in the summary as *providing a layer of abstraction between the presentation layer and the graphical user interface.*

Applicant notes that the "nodes" described in Gipalo are not graphical program nodes as disclosed in the present application, and that Gipalo does not generate or modify the program code itself, but rather modifies data used by the program code to parametrically modify the results of the program execution. As shown in Figure 5C, and described in column 4, lines 6-12, the nodes of Gipalo are data fields used to modify the tree structure (e.g., of user interface screens) *without requiring changes to the underlying code.* In other words, the underlying program code is not modified, merely parameterized by providing data that configures the resulting GUI. Thus, Applicant respectfully submits that Gipalo does not teach graphical program nodes, nor the program generation program of claim 1.

The Office Action further asserts that Volk indicates the limitation *wherein the graphical program comprises a plurality of interconnected nodes that visually indicate* *functionality of the graphical program*, citing column 22, line 58 through column 23, line 20. Applicant respectfully disagrees.

The cited passage describes an interface object inheritance hierarchy, and specifically does not describe nodes in a graphical program as defined in the present application. Applicant notes that such a class hierarchy is not a graphical program, but rather is simply a means for defining objects, in this case, user interface objects. Applicant further notes that a distinction should be made between graphical user interface objects and graphical programs, as they are independent concepts that may be used independently of each other. Thus, as argued above, Volk does not teach or suggest a graphical program. Applicant also notes that Gipalo does not teach or suggest a graphical program, and so even in combination, the cited art does not teach this aspect of Applicant's invention as claimed.

Thus, for at least the reasons provided above, Applicant respectfully submits that neither Volk nor Gipalo, either singly or in combination, teaches or suggests Applicant's invention, and thus, independent claim 1, and claims dependent thereon, are patentably distinct over Volk and Gipalo, and are thus allowable.

Independent claims 29, 30, 44, 45, 46, 56, 65, and 73 contain similar limitations as claim 1, and so for at least the reasons provided above, Applicant submits that these claims, and claims respectively dependent thereon, are also allowable.

Removal of the 103 rejection of claims 1-81 is respectfully requested.

Applicant also asserts that numerous ones of the dependent claims recited further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

## CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-52300/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard<br>
□ Request for Approval of Drawing Changes<br>
□ Notice of Change of Address<br>
□ Check in the amount of $ for fees ( ).<br>
□ Other:

Respectfully submitted,

Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 1/22/2004